

Elisabeth Fughe
Matrikelnummer: 5263769
s3499227@stud.uni-frankfurt.de

Bachelorarbeit (B.Sc. - Informatik)

Können Modelle der Ökonophysik Preis-Dynamiken vorhersagen?

**Fitting von Markt-Modellen auf empirischen Daten mittels Bayesscher
Statistik**

Elisabeth Fughe

Abgabedatum: 16.April 2019

FIAS - Frankfurt Institute for Advanced Studies
Prof. Dr. Nils Bertschinger

Erklärung

gemäß Bachelor-Ordnung Informatik 2011 §25 Abs. 11

Hiermit erkläre ich Frau

Die vorliegende Arbeit habe ich selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst.

Frankfurt am Main, den _____

Unterschrift

Zusammenfassung

tbd

Inhaltsverzeichnis

1	Einleitung	5
2	Verwendete Methoden	6
2.1	Bayessche Modellierung	6
2.2	Markov Chain Monte Carlo (MCMC)	6
2.3	Hamiltonian Monte Carlo Sampling (HMC)	10
2.4	<i>Stan</i> und <i>R</i>	13
3	Die Modelle	17
3.1	GARCH	17
3.2	Vikram & Sinha (VS)	17
3.3	Franke & Westerhoff (FW)	17
3.4	AL herd walk (AL)	17
4	Simulationen	18
4.1	Daten & Vorhersagen	18
4.2	Ergebnisse	18
	Literatur	20

1 Einleitung

tbd

2 Verwendete Methoden

tbd

2.1 Bayessche Modellierung

Die Bayessche Statistik untersucht, mittels Bayesscher Wahrscheinlichkeiten und dem Satz von Bayes, Fragestellungen der Stochastik. Anders als in der klassischen Statistik, die unendlich oft wiederholbare Zufallsexperimente voraussetzt, steht die Verwendung und Modellierung von Wahrscheinlichkeitsverteilungen im Vordergrund.

Es gilt, das beobachtete Daten

$$x = (x_1, \dots, x_n)$$

mittels bedingter Wahrscheinlichkeiten in Beziehung zu unbekannten Parametern

$$\theta = (\theta_1, \dots, \theta_m)$$

stehen. So kann die gemeinsame Wahrscheinlichkeitsdichte

$$p(x, \theta) = p(x|\theta) \cdot p(\theta)$$

durch die a-priori-Verteilung unbekannter Parameter $p(\theta)$ und den Erkenntnissen aus dem Datensatz $p(x|\theta)$ berechnet werden. Mit Hilfes des Satzes von Bayes kann dann die a-posteriori-Verteilung unbekannter Parameter

$$p(\theta|x) = \frac{p(x|\theta) \cdot p(\theta)}{p(x)}$$

ermittelt werden [1].

Die a-posteriori-Verteilung enthält somit Informationen über die unbekannten Parameter, durch die Kombination der a-priori Verteilung mit den Informationen, die aus den beobachteten Daten gewonnen wurden. Sie wird zur Punktschätzung und Schätzung von Konfidenzintervallen genutzt und daher auch oft als Ziel-Verteilung bezeichnet.

So sind Bayessche Modelle, im Gegensatz zur klassischen Statistik, auf kleineren Datensätzen anwendbar, dort ergibt sich jedoch eine breite Wahrscheinlichkeitsverteilung, die somit unter Umständen eine geringe Genauigkeit aufweist.

2.2 Markov Chain Monte Carlo (MCMC)

In der Bayesschen Statistik beschreibt die a-posteriori-Verteilung die Unsicherheit der unbekannten Parameter, die anhand beobachteter Daten geschätzt wurden. Mit der Markov Chain

Monte Carlo (MCMC) Methode kann die a-posteriori-Verteilung und somit die unbekannten Parameter untersucht werden [4].

Um aus der Ziel-Verteilung Stichproben ziehen zu können wird eine Markov-Kette entworfen deren stationäre Verteilung der Ziel-Verteilung entspricht. Hier wird eine Stichprobe aus der a-posteriori-Wahrscheinlichkeitsdichte benötigt, sodass das langfristige Gleichgewicht der Markov-Kette der a-posteriori-Wahrscheinlichkeitsdichte entspricht. Der Zustand der Markov-Kette, er sich möglichst nah am Gleichgewicht befindet, entspricht dann einem Element der Stichprobe. Es gilt somit, dass die Qualität der Stichprobe steigt mit der Anzahl der Schritte, die die Markov-Kette zurücklegt.

Zur Konstruktion der Markov-Kette wird eine Übergangsfunktion berechnet, die die Ziel-Wahrscheinlichkeitsdichte invariant lässt, so dass die Ziel-Dichte der stationären Verteilung der Markov-Kette entspricht. Sei z.B. $p(\theta)$ die Ziel-Dichte und sei $t(\theta'|\theta)$ die Übergangsdichte, dann gilt

$$p(\theta') = \int t(\theta'|\theta) \cdot p(\theta) d\theta$$

Stellt man sich die Markov-Kette als zufällig durch einen Parameterraum wandernden Punkt vor, so entspricht die n -te Komponente der a-posteriori-Verteilung der relative Wahrscheinlichkeit, den Punkt im Zustand n anzutreffen.

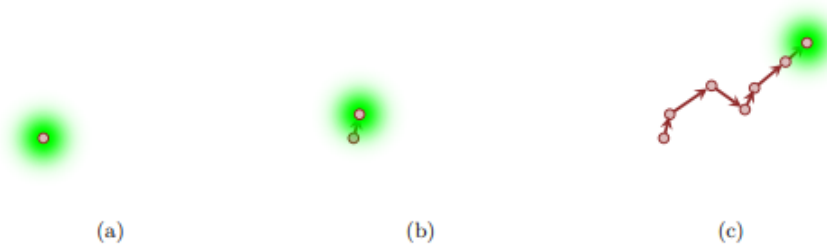


Abbildung 2.1: Verhalten der Markov-Kette beim Ziehen von Stichproben [2]

Die Übergangsdichte(grün) (a) in Abbildung 2.1 beschreibt die Wahrscheinlichkeit eines neuen Punktes im Parameterraum in Abhängigkeit der aktuellen Position. Werden Stichproben aus dieser Verteilung gezogen, also wurden zufällig gewählte nächste Schritte vom Algorithmus akzeptiert, ergibt sich ein weiterer Zustand in der Markov-Kette und eine neue Verteilung von der Stichproben gezogen werden können (b) in Abbildung 2.1. So wandert die Markov-Kette durch den Parameterraum (c) in Abbildung 2.1.

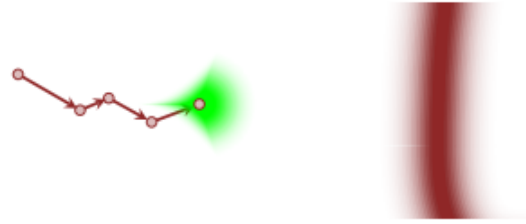


Abbildung 2.2: Konzentration der Markov-Kette auf die typische Menge einer Verteilung [2]

Wenn die Übergangsdichte die Ziel-Verteilung beibehalten soll, konzentriert sie sich auf deren typische Menge (rot) siehe Abbildung 2.2. Unter idealen Bedingungen entdeckt die Markov-Kette in drei verschiedenen Phasen:

1. Die Markov-Kette konvergiert gegen die typische Menge der Verteilung. Die MCMC-Schätzer sind stark verzerrt.
2. Nachdem die Markov-Kette die typische Menge erkannt hat und dort verweilt. Diese erste Erkundung der typischen Menge ist äußerst effektiv und verringert die Verzerrung der MCMC-Schätzer erheblich, da die Verzerrung der ersten Stichproben eliminiert wird.
3. In der dritten Phase wird die Präzision der MCMC-Schätzer durch weiteres Erforschen der typischen Menge weiter verfeinert, so dass sie den zentralen Grenzwertsatz erfüllen. Das bedeutet, dass die durch die Markov-Kette gezogenen Stichproben asymptotisch der Ziel-Verteilung folgen.

Ideal Bedingungen liegen allerdings nicht vor, wenn die typische Menge der a-posteriori-Verteilung z.B. eine starke Krümmung aufweist (siehe Abbildung 2.3). Die Markov-Kette kann damit nicht umgehen und aus diesem stark gekrümmten Bereich keine Stichprobe ziehen und ignoriert diesen einfach und verzerrt dadurch die MCMC-Schätzer. Da Markov-Ketten die exakten Erwartungen asymptotisch wiederherstellen müssen, ist eine Kompensation dafür nötig, dass manche Regionen der Ziel-Verteilung nicht beachtet wurden. Dazu bleibt die Markov-Kette an der Grenze der Krümmung hängen. So werden die MCMC-Schätzer annähernd so berechnet, als würde die Markov-Kette den gekrümmten Bereich erkunden. Allerdings erhält man so überschätzte MCMC-Schätzer. Eine starke Krümmung in der typischen Menge der Ziel-Verteilung führt also zu verzerrten MCMC-Schätzern, die den zentralen Grenzwertsatz nicht mehr erfüllen [2].



Abbildung 2.3: Starke Krümmungen (grün) der typischen Menge einer Verteilung führen zu verzerrten MCMC-Schätzern [2]

Ein weit verbreitetes MCMC-Verfahren zur Konzentration auf die typische Menge der zu untersuchenden Verteilung, ist der Metropolis-Hastings-Algorithmus. Der Algorithmus startet an einem zufälligen Punkt im zu untersuchenden Vektorraum/in der zu untersuchenden Verteilung. Dann wird eine Schrittweite zufällig mit Hilfe einer symmetrischen Wahrscheinlichkeitsverteilung gewählt. Der Schritt wird abgelehnt oder akzeptiert auf Grundlage der Wahrscheinlichkeit der neuen Position im Verhältnis zur alten Position [4]. So wird sicher gestellt, dass die Markov-Kette, von jedem Punkt aus gegen das langfristige Gleichgewicht, der Ziel-Wahrscheinlichkeitsdichte, konvergiert [1].

Der Metropolis-Hastings-Algorithmus ist sehr einfach zu implementieren und liefert gute Ergebnisse, allerdings nicht bei stark korrelierten Parametern [4] und hoher Komplexität der Verteilung [2]. Denn wenn die a-posteriori-Verteilung stark gekrümmt ist, bleibt die Markov-Kette an diesen Rändern hängen und lehnt so gut wie jeden weiteren Schritt ab. So kann nicht aus allen Bereiche der Ziel-Verteilung Stichproben gezogen werden und es entsteht eine verzerrte Stichprobe.

Je mehr Dimensionen die Verteilung hat, desto kleiner wird die typische Menge der Verteilung. Das führt dazu, dass fast immer Punkte außerhalb der typischen Menge gewählt wird. Die Dichte des Punkts ist so klein, dass der Schritt vom Metropolis-Hastings-Algorithmus immer abgelehnt wird. Das führt dazu, dass sich die Markov-Kette kaum bewegt. Erhöht man die Akzeptanz erhöhen, indem man die Anzahl der Punkte reduziert, die sich in der typischen Menge befinden müssen, aber das führt dazu, dass die Markov-Kette extrem langsam konvergiert [2]. In der Theorie würde die Markov-Kette das langfristige Gleichgewicht erreichen, aber in der Praxis stehen dazu nicht unendliche Ressourcen zur Berechnung zur Verfügung. *unnötig?*

Der Metropolis-Hastings-Algorithmus kann auch als stochastisches Optimierungsverfahren verstanden werden. Mittels Simulated Annealing ¹ nähert man sich mit immer höherer Wahrscheinlichkeit an ein Minimum an. Ist die Krümmung der Verteilung zu stark, kann der Algorithmus

¹**Simulated Annealing:** „Ein heuristisches Approximationsverfahren. Es wird zum Auffinden einer Näherungslösung von Optimierungsproblemen eingesetzt, die durch ihre hohe Komplexität das vollständige Ausprobieren aller Möglichkeiten und mathematische Optimierungsverfahren ausschließen“[6].

das lokale Minimum nicht mehr verlassen und so wird fast jeder weitere Schritt der Markov-Kette immer abgelehnt. Letztlich wird der Bereich der Verteilung ignoriert bzw. die Markov-Kette bleibt am Rand des unerwarteten Bereichs hängen. Das führt zu einer Verzerrung der resultierenden MCMC-Schätzer [2].

2.3 Hamiltonian Monte Carlo Sampling (HMC)

In Modellen mit vielen Dimensionen ist es möglich, dass die Markov-Kette mittels Metropolis-Hastings-Algorithmus nicht alle Bereiche der Ziel-Verteilung in einer angemessenen Zeit abtasten kann. Die Strategie des Metropolis-Algorithmus, zufällige Schritte zu wählen und danach zu entscheiden, ob der Schritt akzeptiert wird oder nicht, ist für Modelle mit vielen Dimensionen weniger erfolgreich. Denn es stehen exponentiell viele Richtungen in die sich der Sampler bewegen kann, aber nur ein Bruchteil der Richtungen belässt den Sampler in der typischen Menge und führt letztlich zu akzeptierten Stichproben.

Um also große Schritte vom initialen Punkt in unentdeckte Bereiche der Ziel-Verteilung machen zu können, müssen Informationen über die Geometrie der Verteilung genutzt werden. Die Markov-Kette soll Übergänge machen, die der Masse mit hohen Wahrscheinlichkeiten folgt und sich somit zusammenhängend durch die typische Menge bewegen. Diese Hamiltonian Markov Übergänge können durch Nutzung der differenzierbaren Struktur der a-posteriori-Verteilung berechnet werden.

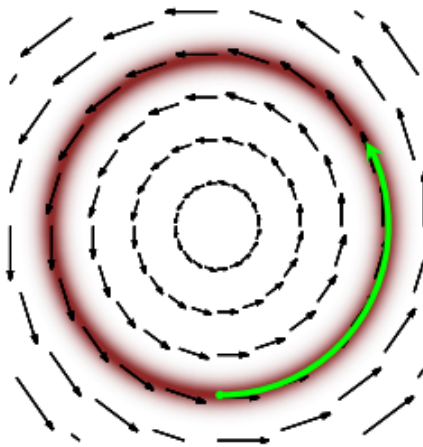


Abbildung 2.4: Mit Hilfe eines Vektorfeldes das an der typischen Menge ausgerichtet ist, kann die Ziel-Verteilung leichter abgetastet werden [2].

Um die Zielverteilung schneller und gezielter abtasten zu können, wird jedem Punkt im Parameterraum eine Richtung für eine kleine Distanz zugewiesen z.B. durch ein Vektorfeld, dass an der typischen Menge ausgerichtet ist (vgl. Abbildung 2.5). Folgt der Sampler

den Richtungen, entsteht eine zusammenhängende Kurve (*engl. Trajectory*), die effizient und möglichst schnell vom initialen Punkt durch unentdeckte Bereiche der typischer Menge der Ziel-Verteilung führt [2].

Um jedem Punkt eine Richtung zuzuweisen, kann ein Vektorraum nur mittels Informationen der Ziel-Verteilung konstruiert werden. Dazu wird die differenzielle Struktur der a-posteriori-Verteilung benötigt, die wir durch den Gradienten der Ziel-Dichte erhalten können. Denn der Gradient der Ziel-Dichte definiert eben ein solches, für die Struktur der Verteilung, sensitives Vektorfeld im Parameterraum.

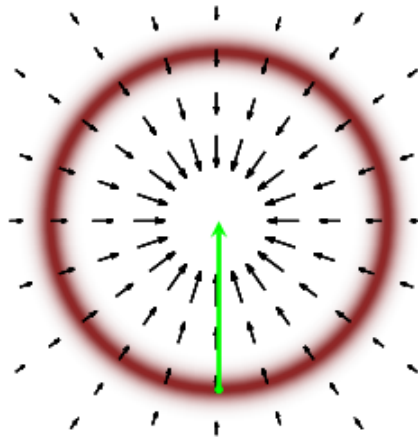


Abbildung 2.5: Die Richtung des Gradienten zeigt immer von der typischen Menge in Richtung parameter-sensitiver Bereiche (grüner Pfeil), wie z.B. dem Modus der Verteilung. Folgt der Sampler diesen Richtungen, kann er die typische Menge nicht erkunden [2].

Allerdings hängen Ziel-Dichte sowie ihr Gradient stark von den Verteilungsparametern ab, sodass der Gradient immer in Richtung parameter-sensitiver Bereiche wie z.B. dem Modus der Verteilung zeigt (vgl. Abbildung ??). Das Vektorfeld enthält somit nicht genügend Informationen, um den Sampler durch die typische Menge in parameter-invariante Bereiche zu führen. Um parameter-invariante Bereiche der Ziel-dichte zu erkunden, muss der Gradient um zusätzliche geometrische Informationen ergänzt werden.

Das ergänzen der Informationen ist durch Differentialgeometrie möglich, die auch die Mathematik der klassischen Physik beschreibt. Es gilt, dass für jedes probabilistische System ein mathematisch äquivalentes physikalisches System existiert [2]. Das Erkunden der Ziel-Dichte kann äquivalent auch als Erkunden eines physikalischen Systems beschrieben werden. Sei der Modus der Ziel-Dichte ein Planet und der Gradient sein Gravitationsfeld, dann sei die typische Menge der Dichte, der Bereich um den Planet, durch den ein Satellit kreisen soll (vgl. Abbildung 2.6).

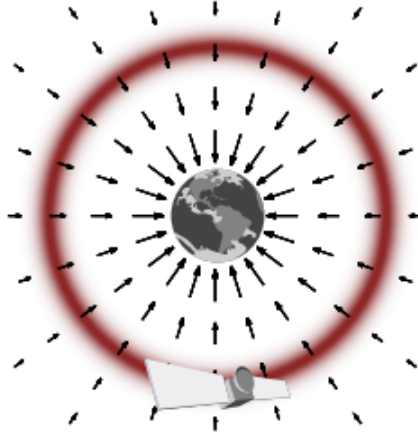


Abbildung 2.6: Analoges System der klassischen Physik [2].

Es gilt also das probabilistisches Modell so um Impulse (*engl. Momentum*) zu erweitern, dass analog der Satellit die Anziehungskraft des Planeten ausbalanciert und die gewünschte Umlaufbahn nicht verlässt [2]. Der Hamiltonian Monte Carlo Algorithmus ist die einzige Algorithmus, der Impulse mit einer probabilistischen Struktur, die konservative Dynamiken erlaubt, implementiert [2].

Der HMC-Algorithmus zieht also Stichproben aus einem erweiterten Zustandsraum mit Impuls i eines Teilchens an der Position P .

Für die Dichte $d(P, i)$ des Zustandsraums gilt:

$$\begin{aligned}
 d(P, i) &= d(P)d(i|P) \\
 &= e^{\log d(P) + \log d(i|P)} \\
 &= e^{-H(P, i)}
 \end{aligned} \tag{2.1}$$

Wobei die Hamiltonian-Korrektur $H(P, i)$, in Analogie zum physikalischen System, der Summe potentieller und kinetischer Energie entspricht.

$$H(P, i) = -\log d(P) - \log d(i|P)$$

Die Hamiltonian Dynamiken \hat{P} und \hat{i}

$$\begin{aligned}
\hat{P} &= \frac{\partial}{\partial i} H(P, i) \\
&= -\frac{\partial}{\partial i} \log d(i|P) \\
\hat{i} &= -\frac{\partial}{\partial P} H(P, i) \\
&= \frac{\partial}{\partial P} \log d(P) + \frac{\partial}{\partial P} \log d(i|P)
\end{aligned} \tag{2.2}$$

erhalten die gesamte Energie bzw. Wahrscheinlichkeit und so kann beim HMC Sampling stets ein Schritt berechnet werden, der akzeptiert wird [1]. Sodass die Markov-Kette auch aus stark gekrümmten Bereichen akzeptierte Stichproben ziehen kann. Besonders in Modellen mit vielen Dimensionen schafft das HMC Sampling so wichtige Effizienzvorteile.

Theoretisch ist HMC Sampling unempfindlich bezüglich stark korrelierter Parameter, allerdings werden die Differenzialgleichungen in der Praxis mittels endlicher Schrittweite gelöst. Das kann dazu führen, dass stark gekrümmte Bereiche der Ziel-Dichte doch nicht mehr erreicht werden. Außerdem gilt es die Schrittweite entsprechend der Parameter Maßeinheiten und -größen anzupassen [1].

2.4 Stan und R

Stan ist eine Open-Source Plattform für statistische Modellierung und high-performance Berechnungen. *Stan* ist für alle in der Datenanalyse weit verbreiteten Sprachen (R, Python, shell, MATLAB, Julia, Stata) verfügbar und läuft auf den gängigen Betriebssystemen (Linux, Mac, Windows) [3].

Stan ist eine höhere Programmiersprache und ermöglicht die Beschreibung der gemeinsamen Verteilung eines Modells auf einer hohen Abstraktionsebene. Die Sprache unterstützt viele Inferenzalgorithmen insbesondere Algorithmen die das Gradientenabstiegsverfahren nutzen, um eine maximale a-posteriori-Schätzung zu erhalten, HMC Sampling sowie Bayesische Gradientenvariation (stochastic gradient variational Bayes). *Stan* Code wird in C++ kompiliert und die hohe Abstraktionsebene der Sprache bietet dem Nutzer einige Vorteile z.B. müssen bei der Nutzung von HMC Sampling die Gradienten nicht manuell implementiert werden. Benötigte Gradienten werden built-in mittels C++ Library für automatisches Differenzieren berechnet werden [1].

Ein *Stan*-Programm muss mindestens einen `data`-, einen `Parameter`-Block sowie einen `model`-Block enthalten. Ein Beispiel dafür ist der *Stan*-Code des in dieser Arbeit verwendeten GARCH-Modells:

Im data-Block, wird definiert welche Daten benötigt werden, um das Modell zu fitten . Die Variablen in diesem Block entsprechen also den tatsächlichen Beobachtungen und der Block definiert, in welchem Format diese Daten vorliegen.

Hier werden zum Beispiel ein Vektor mit Zeitpunkten T, ein Vektor mit Angabe, ob es sich um eine tatsächliche Beobachtung oder um einen zu vorhersagenden Wert handelt, sowie einen Vektor der beobachteten Returns in jedem Zeitpunkt.

Im transformed data Block können berechnete Daten/Variablen definiert werden. Hier zum Beispiel wird über die Daten iteriert und festgestellt, wie viele Einträge als Missing Value markiert wurden.

```
data {
  int <lower=0> T;
  int <lower=0, upper=1> miss_mask[T];
  real ret_obs[T]; // Note: Masked indices treated as missing
}
transformed data {
  int N = 0; // number of missing values
  for (t in 1:T)
    if (miss_mask[t] == 1) N = N + 1;
}
```

Daran anschließend kommt der parameters-Block, der die unbekannten Parameter des Modells definiert. Anschließend daran kann der transformed parameters-Block dazu genutzt werden, um z.B. die Berechnungsvorschrift weiterer Parameter festzulegen.

Hier wird der Return für jeden Zeitpunkt T berechnet: Handelt es sich um eine tatsächliche Beobachtung, wird der beobachtete Return genommen, ansonsten erfolgt die Berechnung mit Hilfe der Standardabweichung im aktuellen Zeitpunkt berechnet:

```
ret[t] = mu + sigma[t] * eps_miss[t]
```

```
parameters {
  real mu;
  real <lower=0> alpha0;
  real <lower=0, upper=1> alpha1;
  real <lower=0, upper=(1-alpha1)> beta1;
  real <lower=0> sigma1;
  real eps_miss[N]; // missing normalized return innovations
}
transformed parameters {
  real ret[T]; // returns ... observed or  $r_t = \mu + \sigma_t * \epsilon_t$ 
  real <lower=0> sigma[T];

  {
```

```

int idx = 1; // missing value index

sigma[1] = sigma1;
if (miss_mask[1] == 1) {
    ret[1] = mu + sigma[1] * eps_miss[idx];
    idx = idx + 1;
} else
    ret[1] = ret_obs[1];

for (t in 2:T) {
    sigma[t] = sqrt(alpha0
                    + alpha1 * pow(ret[t - 1] - mu, 2)
                    + beta1 * pow(sigma[t - 1], 2));
    if (miss_mask[t] == 1) {
        ret[t] = mu + sigma[t] * eps_miss[idx];
        idx = idx + 1;
    } else
        ret[t] = ret_obs[t];
    }
}

```

Zum Schluss kommt der `model`-Block, der die Berechnungsvorschrift der Wahrscheinlichkeitsdichte des Modells enthält [3].

Hier die Annahme, dass μ und σ standardnormalverteilt sind. Die Returns sind normalverteilt zu den Parametern des Modells (μ , σ) wobei es sich, um die Standardabweichung in jedem Zeitpunkt T handelt. Weitere Erläuterungen befinden sich im Kapitel 3.1

Anschließend daran können im `generated quantities`-Block weitere Berechnungen definiert werden. Hier z.B. die Berechnungsvorschrift der bedingten gemeinsamen Wahrscheinlichkeitsdichte.

```

model {
    mu ~ normal(0, 1);
    sigma1 ~ normal(0, 1);

    ret ~ normal(mu, sigma);
    // Jacobian correction for transformed innovations
    for (t in 1:T) {
        if (miss_mask[t] == 1)
            target += log(sigma[t]);
    }
}
generated quantities {

```

```

real log_lik[T];

for (t in 1:T)
  log_lik[t] = normal_lpdf(ret_obs[t] | mu, sigma[t]);
}

```

Der *Stan*-Code der Modelle, die in dieser Arbeit genutzt wurden befindet sich im Anhang und wird im Kapitel 3 näher erläutert.

Die Modelle wurden mittels *Rstan* in *R* gefittet. *Rstan* ermöglicht es *Stan*-Modelle in *R* zu kompilieren, zu testen und zu analysieren. Außerdem wurden die Pakete *tidybayes* und *tidyverse* genutzt, um die Plots zu generieren. Die tidy* Pakete zeichnen sich dadurch aus, dass sie der komplexen Datenstruktur des berechneten Fits, Informationen leicht entnehmen können und in einer Datenstruktur zur Verfügung stellen, die anschließend z.B. mittels *ggplot*-Packet leicht in *R* visualisiert werden kann.

3 Die Modelle

tbd

3.1 GARCH

tbd

3.2 Vikram & Sinha (VS)

tbd maybe

3.3 Franke & Westerhoff (FW)

tbd

random walk - treedepth 128 ?!

3.4 AL herd walk (AL)

tbd

4 Simulationen

tbd

4.1 Daten & Vorhersagen

S&P 500 data in USD from finance.yahoo:

- daily prices - calculated into return and finally into log return as models expect log return as inputs

- exporting data with dates like Jan 1 2000 to Jan 1 2008 yahoo automatically uses last working day at stock exchange

- always used 30 Predictions - inaccurate as days per month change

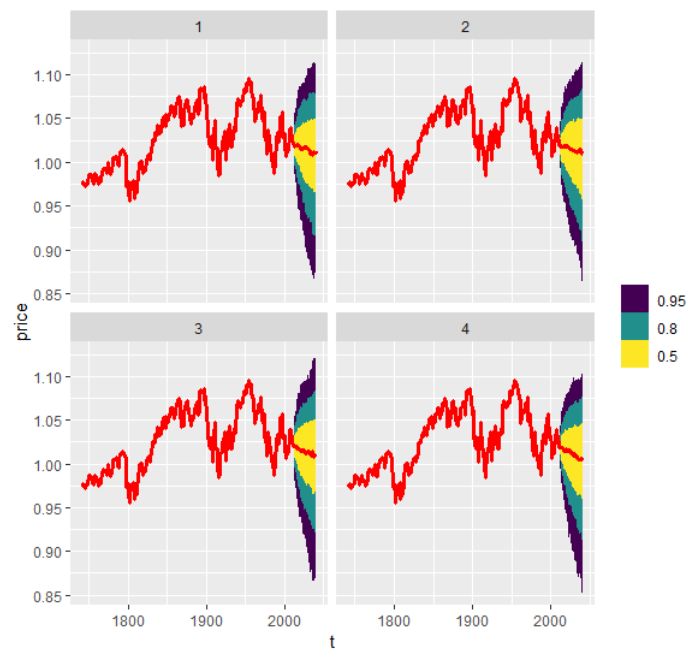
=> this all leads to inaccuracy which is ok as the goal is to see the tendency in which way predictions are shifting

monthly prediction during a year in which a major crisis happend: bank crisis 2008 & dotcom 2000?

used 8 years to predict the year 2008 on a monthly basis: e.g. data from Jan 1 2000 to Jan 1 2008 used to predict Jan 2008

4.2 Ergebnisse

tbd



Literatur

- [1] N. Bertschinger, I. Mozzhorin und S. Sinha. „Reality-check for Econophysics: Likelihood-based fitting of physics-inspired market models to empirical data“. In: *CoRR* abs/1803.03861 (März 2018). arXiv: 1803.03861. URL: <http://arxiv.org/abs/1803.03861>.
- [2] M. Betancourt. „A conceptual introduction to Hamiltonian Monte Carlo“. In: *arXiv pre-print arXiv:1701.02434* (Jan. 2017).
- [3] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li und A. Riddell. „Stan : A Probabilistic Programming Language“. In: *Journal of Statistical Software* 76.1 (Jan. 2017). DOI: 10.18637/jss.v076.i01.
- [4] K. M. Hanson. „Markov chain Monte Carlo posterior sampling with the Hamiltonian method“. In: Bd. 4322. 2001, S. 4322 - 4322 -12. DOI: 10.1117/12.431119. URL: <https://doi.org/10.1117/12.431119>.
- [5] W.S. Kendall, F. Liang und J.S. Wang. *Markov chain Monte Carlo: Innovations and Applications*. Bd. 7. World Scientific Publishing Co Pte Ltd, 2005.
- [6] *Simulated Annealing*. Jan. 2019. URL: https://de.wikipedia.org/wiki/Simulated_Annealing.

Abbildungsverzeichnis

2.1	Verhalten der Markov-Kette beim Ziehen von Stichproben [2]	7
2.2	Konzentration der Markov-Kette auf die typische Menge einer Verteilung [2]	8
2.3	Starke Krümmungen (grün) der typischen Menge einer Verteilung führen zu verzerrten MCMC-Schätzern [2]	9
2.4	Mit Hilfe eines Vektorfeldes das an der typischen Menge ausgerichtet ist, kann die Ziel-Verteilung leichter abgetastet werden [2].	10
2.5	Die Richtung des Gradienten zeigt immer von der typischen Menge in Richtung parameter-sensitiver Bereiche(grüner Pfeil), wie z.B. dem Modus der Verteilung. Folgt der Sampler diesen Richtungen, kann er die typische Menge nicht erkunden [2].	11
2.6	Analoges System der klassischen Physik [2].	12